WHAT IS CLAIMED IS:

1. A recommender for usage in a data mining system comprising:

a scoring engine capable of scoring information using at least one data mining model that derives scores from aggregated data;

an offer manager capable of mapping scores to offers based on entered configuration information;

an aggregation engine that computes user-defined aggregates dynamically for real-time scoring by the scoring engine;

a user interface that exposes entity data through a user-written Interface Device Language (IDL), the IDL defining a context object containing the input data including record-sets, singular records, and scalar elements; and

at least one aggregate wizard that defines aggregates and reflects on the user interface.

2. The recommender according to Claim 1 wherein:

the aggregation engine is capable of computing aggregates in real-time from in-memory data that is dynamically cached during a session with an entity.

3. The recommender according to Claim 2 further comprising:

an aggregation wizard that enables a user to select a record collection in the in-memory data, select an element of the record collection as a value to be aggregated, select an aggregate function, select a condition to apply, and aggregate by applying the selected condition.

4. The recommender according to Claim 3 further comprising:

an aggregation wizard that enables a user to select an element upon which the condition is based, select an operator, and select a value for comparison.

5. The recommender according to Claim 3 further comprising:

an aggregation wizard that enables a user to write a custom static method to compute derived attributes.

6. An information handling method comprising:

constructing aggregates for a case set;

computing aggregates from detailed in-memory data that is dynamically cached; and

iteratively mining data during an entity transaction to define new aggregates and refine existing aggregates based on entity responses, the aggregates being defined and refined using a wizard.

7. The method according to Claim 6 further comprising:

performing aggregation for real-time scoring.

8. The method according to Claim 6 further comprising:

drawing aggregates from different tables in a memory containing entity-related data.

9. The method according to Claim 6 further comprising:

computing a compound aggregate from at least one other aggregate.

10. A real-time information handling apparatus comprising:

an interaction manager that populates a recommender context with data cached for an entity transaction session;

a recommender that receives the recommender context, the recommender further comprising:

at least one recommender driver and/or utility;

a custom recommender server that is configured via the at least one recommender driver and/or utility dynamically loading a stub; and

an aggregation engine defined using a graphical user interface wizard that reflects upon Java classes generated from the graphical user interface to identify available records and attributes.

11. The apparatus according to Claim 10 wherein the recommender further comprises:

a scoring engine capable of computing scores for at least one predefined business model and to make offers contingent on resulting scores.

12. The apparatus according to Claim 10 wherein the recommender further comprises:

an offer manager capable of using criteria entered into a table to select offers based on scores from at least one data mining model.

13. The apparatus according to Claim 10 wherein the recommender further comprises:

an interface capable of reloading and compiling a combination of metadata categories in real-time including aggregate definition models, deployed data mining models, business rules, and offer definitions.

14. The apparatus according to Claim 10 wherein:

the recommender context comprises stubs and skeletons, and structures and/or record-sets, a distinct class being generated for each structure or record set, the class representing the recommender context.

15. An information handling method comprising:

populating a recommender context with data cached for an entity transaction session;

compiling a customized interface definition language file to generate stubs and skeletons;

generating a distinct class for each structure or record-set including each record-set as an array of objects;

dynamically loading a skeleton to create a custom server;

defining aggregates using a wizard that reflects upon classes generated from an interface to identify available records and attributes; and

computing aggregates from data cached in memory.

16. An information handling method comprising:

creating simple aggregates using a simple aggregate wizard comprising:

selecting a record-set to be aggregated from a list populated with all objects in a context class;

selecting an element for aggregation from a list populated with fields of an object of the context class objects; and

selecting an aggregation function from a list populated with known aggregate functions applicable to a data-type of a field of the object fields.

17. The method according to Claim 16 further comprising:

selecting zero or more conditions for the aggregate, a condition capable of being autonomously selected based on the element selection.

18. The method according to Claim 16 wherein:

the aggregation functions include at least one function selected from among a group consisting of *count*, *sum*, *min*, *max*, *mean*, *count-distinct*, *and*, *or*, *first*, and *last*, the functions including arithmetic functions and Boolean functions.

19. The method according to Claim 16 further comprising:

defining a context to include a struct to designate a unique record or a record containing pre-computed historical aggregates.

20. The method according to Claim 16 further comprising:

aggregating an element that is a column of a record in a first case; and

aggregating an element by deriving an attribute from operations on a plurality of at least one column of the record in a second case.

21. An information handling apparatus comprising:

an aggregation wizard further comprising:

a first list populated by elements of structure and by a derived method that takes class of the structure as a parameter, the first list alternatively listing items including: a first item type of static methods that can be written by an entity to compute a derived attribute of a record, and a second item type of a class that matches class of a static method parameter; and

a second list populated by types of elements, the second list being operative for the second item type and capable of listing static methods capable of attribute form conversions.

22. The apparatus according to Claim 21 wherein:
attribute form conversions are selected from among a group comprising a parse, an extraction, a mathematic operation, and a Boolean operation.

23. The apparatus according to Claim 21 wherein:
for the second item type, the second list is populated upon selection of a conversion method with a new set of conversions relevant to a data type returned by the most recent conversion method selected, the data types being selected from among a group comprising a primitive data type, a string, a standard class, or an entity-written class.

24. The apparatus according to Claim 23 wherein:
for conversion to a class type, the second list displays applicable static methods, public fields, and derived fields.

25. The apparatus according to Claim 21 further comprising:
a relational operator static method for which the aggregation wizard:
parses a string parameter and returns an object of a relational operator class;
operative when the second list includes a relational operator, displaying fields and get-methods of the relational operator class; and
extracting a corresponding value for comparing against the relational operator.

26. The apparatus according to Claim 25 further comprising:
a relative element static method for which the aggregation wizard:
receives a relational element parameter and returns a number relative to an origin, designated as positive and negative relative to the origin.

27. The apparatus according to Claim 21 wherein:

the function page enables the user can enter a name of a custom aggregate.

28. The apparatus according to Claim 21 wherein:
the aggregation wizard dynamically loads and verifies a user-written class for each known data type.

29. An information handling apparatus comprising:
a compound aggregation wizard further comprising:
a function selection page that enables a user to select a compound aggregate function from a list, the function determining component aggregate number and types;
a component page that enables the user to sequentially select at least one component aggregate from a list populated with all currently defined aggregates of an applicable type; and
a compute method that computes the compound aggregate from the component aggregates.

30. The apparatus according to Claim 29 wherein:
the component page displays a finish button that enables a user to complete selection of the component aggregates, the number of component aggregates for a particular compound aggregate being fixed for some aggregates and variable for some aggregates.

31. The apparatus according to Claim 29 wherein:
the function page enables the user to define a name of a custom compound aggregate class.

32. The apparatus according to Claim 29 wherein:
the compound aggregation wizard dereferences fields and methods, performs conversions, and selects applicable rows of a data array.

33. An article of manufacture comprising:

a controller usable medium having a computable readable program code embodied therein for mining data, the computable readable program code further comprising:

a code capable of causing the controller to populate a recommender context with data cached for an entity transaction session;

a code capable of causing the controller to compile a customized interface definition language file to generate stubs and skeletons;

a code capable of causing the controller to generate a distinct class for each structure or record-set including each record-set as an array of objects;

a code capable of causing the controller to dynamically load a skeleton to create a custom server; and

a code capable of causing the controller to define aggregates using a wizard that reflects upon classes generated from an interface to identify available records and attributes.

34. An article of manufacture comprising:

a controller usable medium having a computable readable program code embodied therein for mining data, the computable readable program code further comprising:

a code capable of causing the controller to create simple aggregates using a simple aggregate wizard comprising:

a code capable of causing the controller to select a record-set to be aggregated from a list populated with all objects in a context class;

a code capable of causing the controller to select an element for aggregation from a list populated with fields of an object of the context class objects; and

a code capable of causing the controller to select an aggregation function from a list populated with known aggregate functions applicable to a data-type of a field of the object fields.